Designing a Robotic Platform for Investigating Swarm Robotics

Jonathan Gray

A Senior Thesis submitted in partial fulfillment
of the requirements for graduation
in the Honors Program
Liberty University
Spring 2019

Acceptance of Senior Honors Thesis

This Senior Honors Thesis is accepted in partial fulfillment of the requirements for graduation from the Honors Program of Liberty University.

____
Kyung Bae, Ph.D.
Thesis Chair

____
Feng Wang, Ph.D.
Committee Member

____
Daniel Majcherek, Ph.D.
Committee Member

____
David Schweitzer, Ph.D.
Assistant Honors Director

____
Date

Abstract

This paper documents the design and subsequent construction of a low-cost, flexible robotic platform for swarm robotics research, and the selection of appropriate swarm algorithms for the implementation of a swarm focused predominantly on target location. The design described herein is intended to allow for the construction of robots large enough to meaningfully interact with their environment while maintaining a low per-robot cost of materials and a low assembly time. The design process is separated into three stages: mechanical design, electrical design, and software design. All major design components are described in detail under the appropriate design section. The BOM for a single robot is also included, along with relevant testing information.

Designing a Robotic Platform for Investigating Swarm Robotics

Introduction

Introduction to Swarm Intelligence

Swarm intelligence is decentralized intelligence – a collective intelligence that arises in a group of similar organisms. Unlike a standard hierarchical control structure, a swarm has no ranks or concepts of authority. All agents of a swarm are of equal importance and possess equivalent capabilities. Agents composing a swarm are not required to possess great intellectual capabilities, as the strength of a swarm is very much in its numbers. Upon the gathering of a significant enough number of similar organisms following the same basic rules into a swarm, a group intelligence arises that governs the swarm's actions, guiding the swarm with an intelligence much greater than the individual members of the swarm could offer. This apparent group organism allows swarms to accomplish remarkable tasks that should not be possible for organisms of the swarm members' complexity (or lack thereof).

There are many examples of swarm intelligence in the natural world, some more effective than others. Both colonies of ants and swarms of bees exhibit a swarm intelligence adept at finding the most efficient ways to collect food and return it to their colonies, though both use different methods to accomplish the same goals. Other insects, such as termites and wasps, can construct complicated and enormous nests for their colonies, even though none of the participating members of the swarm have any kind of master plan or higher intelligence. Individually, these organisms are more or less helpless, but together they can accomplish significant tasks (Blum & Merkle, 2008).

Due to the ability of swarm intelligence to allow a group of simple agents to complete complex tasks, it has many possible applications for humans. In order for this intelligence to be utilized effectively, algorithms must be created that mimic the actions of a biological swarm. These swarm algorithms are based on the swarm behavior of many different organisms, including ants (Ant Colony Optimization, ACO), fish and birds (Particle Swarm Optimization, PSO), bees (Artificial Bee Colony, ABC), and glowworms (Glowworm Swarm Optimization, GSO) (Ab Wahab, Nefti-Meziani, & Atyabi, 2015). These algorithms, in their most basic form, are a set of simple rules that every agent in the swarm follows. Due to the ability of swarm intelligence to allow a group of simple organisms to function with a much higher level of intelligence than they are individually capable of, swarm algorithms have many technological applications, including the interesting world of swarm robotics.

Introduction to Swarm Robotics

Swarm robotics is the application of swarm intelligence to robotic systems. Like biological swarms, robotic swarms are composed of many agents with equivalent abilities and significantly lower complexity than what should be required to complete the allocated task. This allows swarms of simple robots governed by swarm algorithms to complete complicated tasks efficiently. Robotic swarms also mimic biological swarms' ability to share information across the swarm; unlike biological swarms, however, robotic swarms can make use of wireless communication technology to instantly exchange information with every other agent in the swarm, allowing for much greater flexibility in swarm implementations. The method of communication, types of sensors, swarm algorithms used, and other functional aspects of the swarm members can be adjusted to allow for an optimized swarm implementation for the given task.

Purpose of Project

Swarm robotics is a relatively new area of research, and as such there is still plenty to learn about applying swarm algorithms to robotic systems. One limiting factor to such research is the fact that to research swarm robotics, one should ideally have a physical swarm, as simulations do not always address some areas of interest. Creating a swarm can be an expensive and time-consuming endeavor, which limits the research that can be conducted. One example of swarm robotics research is Harvard’s Kilobot project, a swarm of a thousand simple robots for investigating swarm behavior in large groups (Perry, 2014). To reduce the price and time required to construct their swarm, the researchers involved in the project reduced the abilities of each robot to the point that each robot has no practical purpose individually. While this approach is functional when one is investigating swarm algorithms, it is not practical for experimenting with a swarm that is intended to be capable of interacting with its environment in any notable way (e.g., traveling long distances, moving objects, gathering multiple kinds of sensory data).

This project is intended to address this issue by creating a reliable, functional platform for swarm robotics research, with a focus on target acquisition. In order to successfully accomplish this, each robot should be able to be constructed for a reasonable price and with minimal time requirements. Additionally, each robot should have the features and functionalities listed below:

- Basic navigation capabilities: Each robot should have the ability to know its approximate location in relation to the point it started from, including distance traveled. This data should be accurate enough that other robots in the swarm can use it to travel to the first robot's position.
- Ranged communication abilities: Each robot should possess a wireless communication method that enables communication with other swarm members out of line of sight (i.e., around obstacles, not necessarily long distances). Additionally, a secondary line of sight method for short-range communication would be ideal.
- Obstacle detection: Each robot should have the ability to detect obstacles in its path and be able to measure the distance to the object.
- Target detection: As the focus of this swarm is target acquisition, each robot should have the ability to detect a target and distinguish it from other obstacles in some way. This could be implemented using the short-range communication listed above.
- Reasonably long battery charge: Each robot should be able to operate at full capacity on a fully charged battery for a reasonable amount of time (at least 20 minutes). This allows for meaningful swarm investigations to be carried out with minimal interference.
- Ability to accept modifications: Each robot should be able to carry a reasonable amount of extra weight (at least 250 g). This allows for the addition of customized

sensors or other modifications, increasing the capabilities and versatility of the swarm.

- Ability to run swarm algorithms: Each robot must possess the ability to execute swarm algorithms to govern its behavior and interactions with other members of the swarm. To accomplish this, each robot must be controlled by a microcontroller with a sufficiently high clock speed, enough RAM, and enough EEPROM.

In addition to the physical requirements for the swarm, there are requirements that must be met by the selected algorithms. As noted, the focus of this swarm is intended to be target acquisition. The selected algorithm must therefore provide the following abilities:

- Swarm convergence: The selected algorithm should cause the swarm to converge on the target, allowing each robot to alter its course based on data gathered by other robots in the swarm.
- Route optimization: The selected algorithm should select the most optimized route from each member of the swarm to the target, once it has been discovered.

A robot that can meet all the above hardware and software requirements, while maintaining a low required investment of time and money, has the potential to be a powerful tool for investigating applications of swarm robotics to real-world problems.

Building a Swarm

The design of the swarm consists of three main stages: mechanical design, electrical design, and software design. The functionality of the swarm will be dependent upon the quality of each of these design stages. The mechanical design stage occurred first, as it determines what options are and are not feasible with respect to electrical and software design.

Mechanical Design

The mechanical design stage covers the design of the skeleton of the swarm robots and was the starting point in the design process. This section addresses the selection of the method of locomotion, type of motors used, size and shape of the swarm robots, and the type of material used to construct the bodies of the robots.

Methods of locomotion. The first design issue considered was locomotion. There are three main options for locomotion when it comes to a ground-based swarm: wheels, continuous track (tank treads), or legs. Legs were quickly ruled out for this swarm implementation since their increased complexity and cost outweigh their increased maneuverability over rough terrain. Continuous track was considered as a viable option, as it increases the ability of a robot to handle rough terrain while also reducing the chance of slippage. Unfortunately, using continuous track for each robot in the swarm would either require enough money or increased assembly time as to render the idea impractical. The remaining option, wheels, was the selected option, due to wheels being low cost enough to be practical while also having a high grip to reduce slippage.

Physical dimensions. The second issue considered was the desired size and shape of the swarm robots. As discussed earlier, one of the goals of this project was to create a design that was both large enough to meaningfully interact with the environment while still requiring a relatively small investment of time and money to construct. Each robot should also be small enough that storing a swarm can be easily achieved. As such, the diameter of each robot was limited to eight inches or less, with a six-inch diameter being the final selected value. Two main general shapes were considered for the base of the robots: a circular base and a rectangular base. In testing, the rectangular base was shown to have trouble navigating in tight areas, due to its corners becoming snagged on obstacles when maneuvering. The circular base performed much better in navigation tests, as it could rotate fully without moving in any direction, thus allowing it to navigate through more difficult terrain than the rectangular base. Due to this notable advantage, the circular base was selected for the swarm robot platform.

Selection of motors. Motors are often one of the most expensive parts of a robot, but at the same time high maneuverability and velocity are important factors for a swarm focused on target location, so effective motors must be used. For a small robotic platform, there are four main options: continuous rotation servos, stepper motors, brushless DC, and brushed DC. Continuous rotation servos have a built-in gearbox that reduces speed in exchange for torque, which is a necessary function for small motors. Additionally, continuous rotation servos provide feedback to the microcontroller driving them, allowing the microcontroller to control how many degrees the servo rotates. This is a very useful feature for a motor to possess, as keeping track of the distance a robot has traveled is essential to the operation of the swarm. As a downside, however, continuous rotation servos are significantly more expensive than many other types of motors, and as such were not suitable for this project. Stepper motors are another type of motor that allows for precise rotation control, but also have a high enough price tag that using them for swarm applications is not practical. Additionally, stepper motors require extra control circuitry to drive each of the motor's phases properly. Brushless DC motors were the third option considered: they are all-around more efficient than brushed DC motors, packing more power into a smaller package, but again, the increased cost and extra control circuitry prevented BLDC motors from being a viable option. The final option, brushed DC motors, was selected due to their low price, ease of use, and minimal external control circuitry. To increase the torque of the motor, an external gearbox was fastened to each motor. Even with this addition, the price of the motor setup for the swarm robots still came to a much lower price than a comparable configuration utilizing any of the other available options would have cost. One issue that must be considered in this choice is that the selected motors have no means of tracking the distance traveled, so that must be addressed in electrical design.

Selection of material. The final mechanical design consideration was the material from which to create the base of the swarm robots. The selected material must be low-cost, rigid, and light enough that it does not strain the motors. For this design phase, there were three considered materials: plywood, 3D printed ABS, and hardboard (HDF). All three options are sturdy, rigid materials, and none of them have a high price tag. The 3D printed ABS was the first option eliminated, simply due to the large amount of time that would be required to print any significant number of the required base pieces. The plywood was more durable than an equivalent thickness of HDF, but also somewhat more expensive. The hardboard was eventually selected over the plywood, as it was determined to be sturdy enough to fulfill its intended purpose at a lower cost.

Overall mechanical design. Based on these design decisions, an overall mechanical design was drafted. The body of the swarm robot is composed of two layers, each a six-inch diameter circular shape cut from HDF. The designs of the top and bottom layers, designed in Adobe Illustrator CC, are shown in Figures 1 and 2.

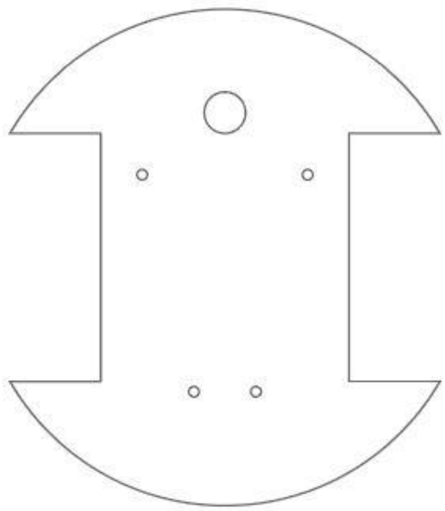

Figure 1. Swarm robot upper layer

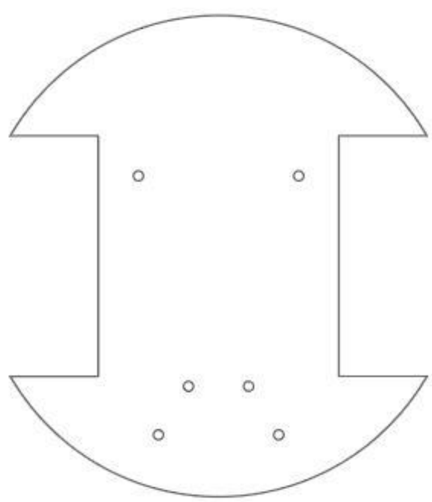

Figure 2. Swarm robot lower layer

The DC brushed motors and gearboxes are sandwiched between the hardboard layers, with the wheels being mounted to the gearboxes in cutouts on each side of the base. Since a circular body design was used, only two wheels could realistically be mounted, so a flange-mounted ball bearing was used as a third point of contact to stabilize the robot (see Figure 4). Pictures of a prototype swarm robot body are shown in Figures 3, 4, 5, and 6.

Figure 3. Top view

Figure 4. Bottom view

Figure 5. Profile view with wheel

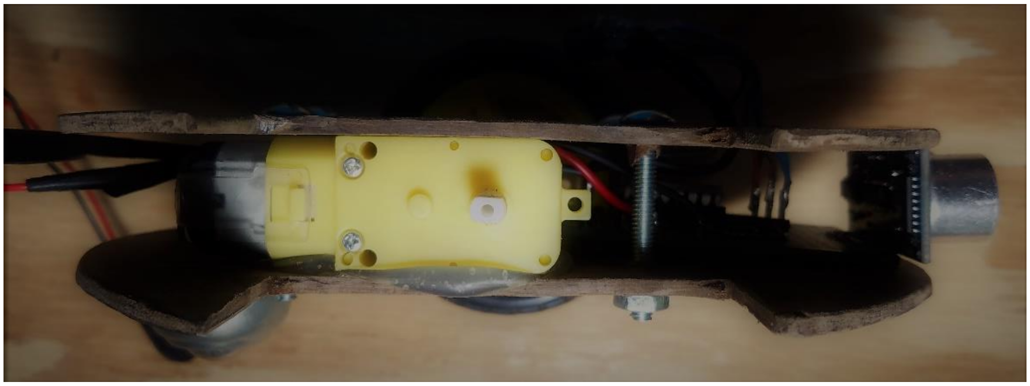

Figure 6. Profile view without wheel

Electrical Design

The electrical design phase covered all the sensor circuitry, control circuitry, communications circuitry and power supply, and was based in part upon the mechanical design phase (i.e., all circuitry must properly fit in or on the body of the swarm robot).

Selection of sensors. The base sensors required for a functional target-location-oriented swarm implementation are some form of range finding sensor and a sensor that can distinguish between obstacles, other members of the swarm, and the desired target. Unless powerful microcontrollers and machine vision are used, range-finding and target detection sensors are limited to those which transmit a signal and detect the reflected signal to determine if an object is in the path of the swarm robot. Realistically, there are two main options for these types of sensors: infrared (IR) and ultrasonic. Due to the speed of sound through air being very slow in relation to a microcontroller's clock speed, ultrasonic sensors allow the delay in signal reflection to be used to determine the distance to the obstacle, while IR sensors must simply use the amplitude of the reflected signal to provide a rough estimate of range. This fact makes ultrasonic sensors more accurate for range finding and obstacle detection, although they can be augmented with IR sensors for improved performance, as ultrasonic sensors sometimes have issues detecting soft materials and sharp corners. IR sensors, on the other hand, allow for relatively straightforward data transmission to line of sight targets, making them ideal for differentiating another swarm robot or swarm target from an environmental obstacle. Each robot in the swarm (and the target) can be configured to transmit a unique identification code via their IR transmitters; whenever another swarm robot detects that code propagating from a detected obstacle, it knows that the obstacle is another robot in the swarm. Additionally, the robot will know exactly which member of the swarm it has encountered, which will allow it to check its own recorded position against the detected robot's recorded position and correct for any errors that may have accumulated over time. This method could also be used to perform a rough triangulation, allowing swarm robots to more accurately calculate the position of other robots in relation to themselves. Due to these factors, an ultrasonic range finder was selected for obstacle detection and an assortment of IR emitters and detectors were selected for obstacle discrimination and line of sight communications.

In addition to these two necessary sensors, an assortment of other non-necessary sensors was considered, including sonic (microphone), thermal, color, and light intensity sensors. The microphones would be accompanied by basic audio processing circuitry such as filters, amplifiers, and decoupling capacitors, which would then pass the processed audio signal to the microcontroller. The microcontroller could measure the frequency and amplitude of the audio signal, which would allow swarm experiments to introduce audio as an environmental variable. This could be used experimentally to test the application of swarm robotics to identifying the source of a particular signal, which would be quite practical in a swarm intended for experimenting with target location.

Simple thermal sensors, such as thermistors, provide two benefits to the swarm robot: first, they provide another experimental variable, similar to audio as discussed above; and secondly, they provide a means to continuously stabilize the accuracy of the ultrasonic range finder, as the speed of sound through air is based on the temperature of the air. Having access to a thermal sensor allows the swarm robot to constantly ensure that the most accurate value for the speed of sound is being used in range finding calculations, increasing the swarm’s performance. Additionally, having an on-board thermal sensor allows the swarm to create thermal maps of an area, which could be useful experimental data. If a thermistor is used as the thermal sensor, the only necessary support circuitry would be a single resistor to form a voltage divider, the output of which could be sent to the microcontroller through an ADC.

Another optional sensor considered was a color sensor. In its simplest form, a reliable color sensor would consist of a red-green-blue (RGB) LED and a phototransistor. To detect the color of a surface, the microcontroller would record the output of the phototransistor while alternately flashing the RGB LED red, green, and blue. Based on the phototransistor outputs at each color, the robot could determine the approximate color of the surface the sensor is pointed at. To increase the accuracy of the results, light from other sources should be minimized at the sensor. To achieve this, the sensor can be placed on the underside of the robot, facing directly at the ground. This arrangement allows the swarm robot to detect the color of the surface it is traveling over, providing yet another test variable for swarm experiments. Just like in the case of the thermal sensor, a swarm equipped with color sensors could create a colormap of the area they cover, which could provide a secondary comparison value to allow swarm members to fine-tune their estimated positions.

Light intensity sensors, such as photoresistors, would provide the swarm with a way to detect changes in intensity of visible light. This could be recorded by each robot in the swarm and used to create a shared map of light intensities across the area covered by the swarm. A light intensity map would effectively be a map of shadows across the area of investigation and could be an additional positional verification tool to allow swarm robots to corroborate their locations. If combined with an LED, an on-board photoresistor could also be used for frequency modulated VLC experiments. The support circuitry for a photoresistor would consist of only a single resistor to form a voltage divider, the output of which would be passed to the microcontroller through an ADC. If desired, a pair of photoresistors could be used, one placed on each side of the front of the swarm robot. This would allow the robot to immediately determine from which direction the light was coming, which could be a helpful ability, depending on the experiment being carried out.

As noted in the introduction, two of the objectives of this project are to design a swarm that is both low-cost and able to accept modifications. In the interest of keeping the cost low, the sensors which will be included in the basic design will be the ultrasonic range finding sensor, an IR emitter/detector pair, and a thermistor to allow for more accurate distance measurements from the ultrasonic sensor. To allow for easy modification and interfacing of additional sensors, the microcontroller and required support circuitry will be placed in an easy to access manner.

Selection of wireless communications. For a group of robots to function as a swarm, they must be able to communicate. For a swarm focused on target location, each robot should be able to communicate with any other robot in the swarm, even if the two robots are not within line of sight of each other. These requirements mean that an RF communications method must be selected. The two main RF communications options considered were a Bluetooth module (HC-06 specifically) and an ISM band transceiver (nRF24L01+). Both have relatively similar performance, but the nRF24L01+ was the cheaper option, so it was selected for this project. As noted above, the nRF24L01+ is an ISM band transceiver operating at 2.4 GHz. It supports an SPI connection for data transfer between the radio and a microcontroller, making it easy to use. GFSK modulation is used, and the transceiver supports the use of addresses, so each robot in the swarm can have a unique address assigned to it. Additionally, the nRF24L01+ supports up to 2 Mbps data transfer rates, allowing robots in the swarm to transfer relatively large amounts of data between themselves (Nordic Semiconductor, 2008).

Selection of microcontroller. In the interests of reducing assembly time, reducing project cost, and supporting selected sensors, it was decided that a pre-assembled microcontroller board (as opposed to an independent IC) would be used. Since several of the sensors for the swarm robots require an ADC, the selected microcontroller board should have an ADC built in. The selected board should also have the required RAM, EEPROM, and clock speed to successfully handle the required processing for executing the swarm algorithms and all subroutines. After researching qualifying options, an Arduino Nano clone was selected, mostly due to the price. The selected board provides a 16 MHz clock, 32 KB flash memory, 2 KB SRAM, an eight channel 10b ADC, and is based on the ATmega328 microcontroller (“Arduino”, n.d.). Based on these features, it should be capable of performing well in the scope of this project.

Selection of power supply. The power supply selection is governed by several constraints: first, it must be rechargeable in order for it to be a feasible option; secondly, it must fit in the space between the upper and lower sections of the robot body, to leave the upper layer clear for circuitry; thirdly, it must supply at least seven volts under load; and finally, it must be able to supply up to one amp constantly, with spikes up to 1.5 amps. Based on these limitations, the three considered options were lithium-ion batteries, a rechargeable 9V battery, and rechargeable AAA batteries. For the purposes of this project, 9V batteries were used to power the swarm robots due to their small size and standard terminal. Since the selected microcontroller board has an on-board voltage regulator, the battery could be replaced with any other battery that meets the size requirements, provided it supplies between seven and twelve volts.

Design of wheel rotation tracking system. Since the selected motors do not include any form of rotation tracking, some external solution must be implemented in order to keep track of the distance traveled by the robot for position calculations. Due to the limited space, rotary encoders are not really practical, so a simple IR solution was selected: black and white striped disks can be glued to the inside rim of the wheels, while an IR emitter and receiver pair face the disk. The IR LED illuminates the surface of the disk, reflecting off the white stripes and being absorbed by the black stripes. The IR receiver detects the reflected light and passes the signal through a Schmitt trigger to clean it up and convert it to a square wave. The output of the Schmitt trigger is passed to the microcontroller, which can then count the number of rotations or fractions of rotations and store the data for position calculations. This circuit would be implemented for both wheels, so the robot would also know approximately how many degrees it rotates while turning. Of course, errors in measurement will stack up, throwing off the accuracy of such a system, but the intended purpose of this system is simply to allow the robot to navigate close enough to its target, whereupon its other sensors, especially the IR short range communication, will allow it to locate its target.

Design of motor control circuitry. Since the mechanical design for the swarm robots allows a robot to rotate in place, the ability to drive in reverse is not absolutely necessary, but it does allow for greater maneuverability for the swarm. Since two brushed DC motors are used, two H-bridge circuits will be needed to allow for forwards/backwards drive capability. To simplify construction, a dual channel H-bridge IC (L293D) was selected for motor control. The L293D does not require any support circuitry, so it makes for a simple and cost-effective solution to the motor control issue.

Overall electrical design. The overall electrical design of a swarm robot consists of an Arduino Nano clone as the microcontroller, with an nRF24L01+ 2.4 GHz ISM transceiver for communication, IR emitter/detector pairs for line of sight communication, an ultrasonic sensor paired with a thermistor for range finding and distance calculation correction, an IR wheel rotation tracker, and an L293D motor driver, with power supplied by a 9V battery. Schematics and PCB designs (ground planes not shown for clarity) for the IR wheel rotation sensor and motor control are shown in Figures 7 through 10.

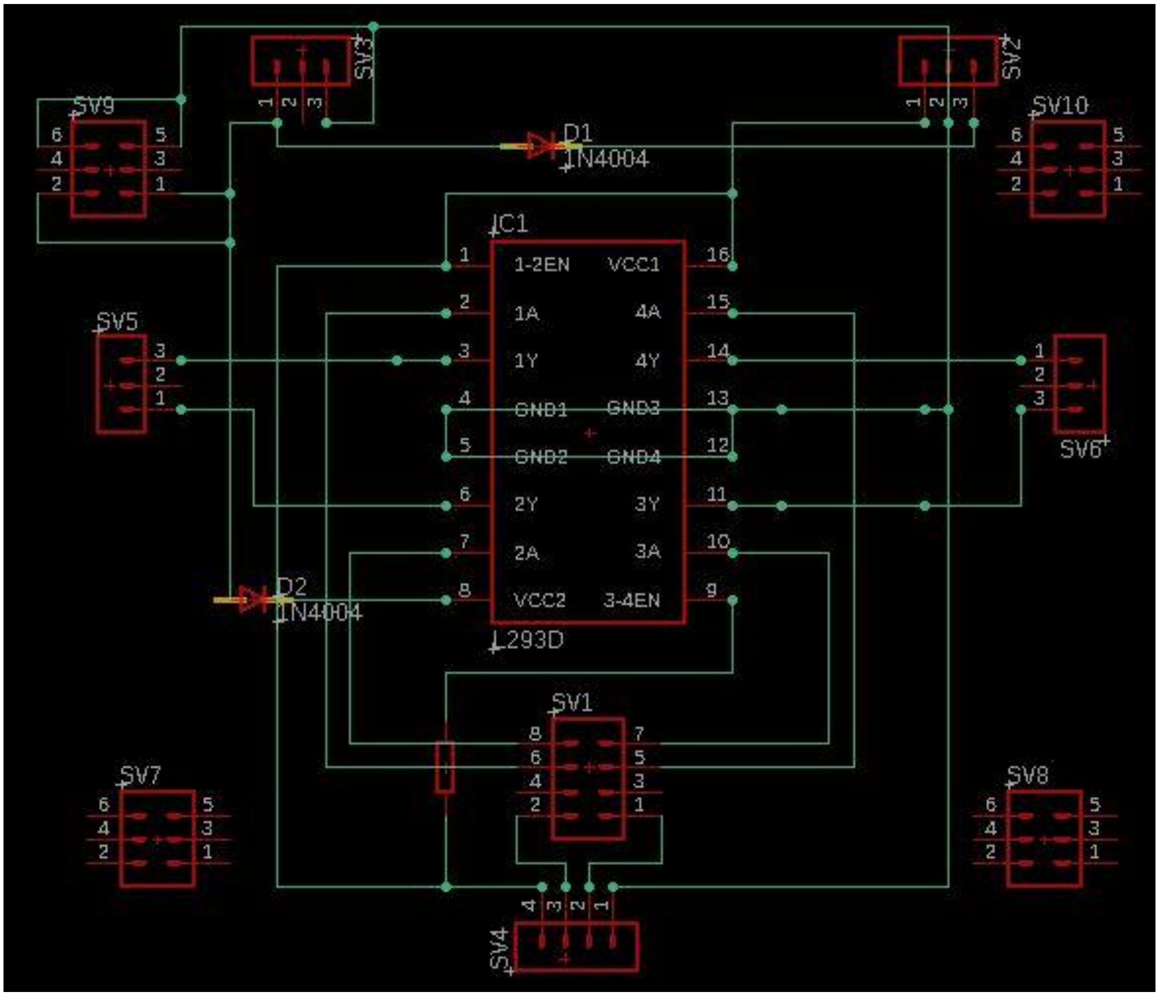

Figure 7. motor control schematic

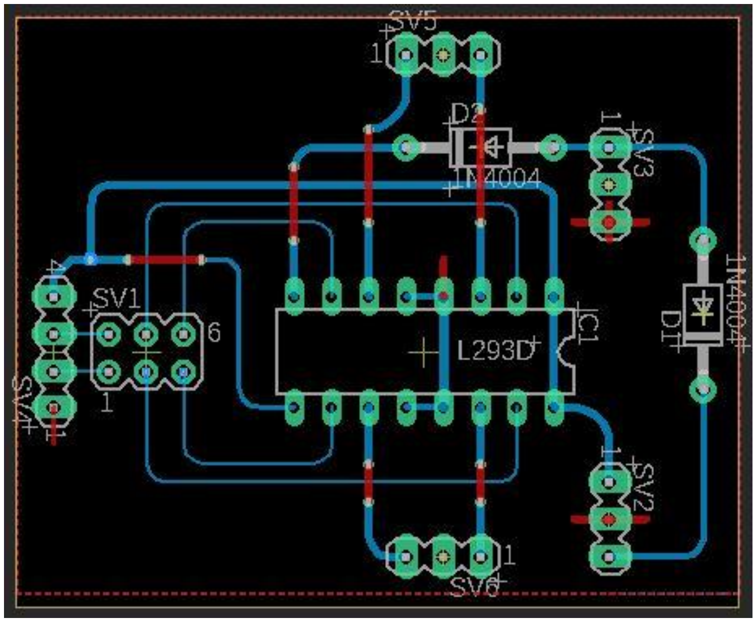

Figure 8. motor control PCB

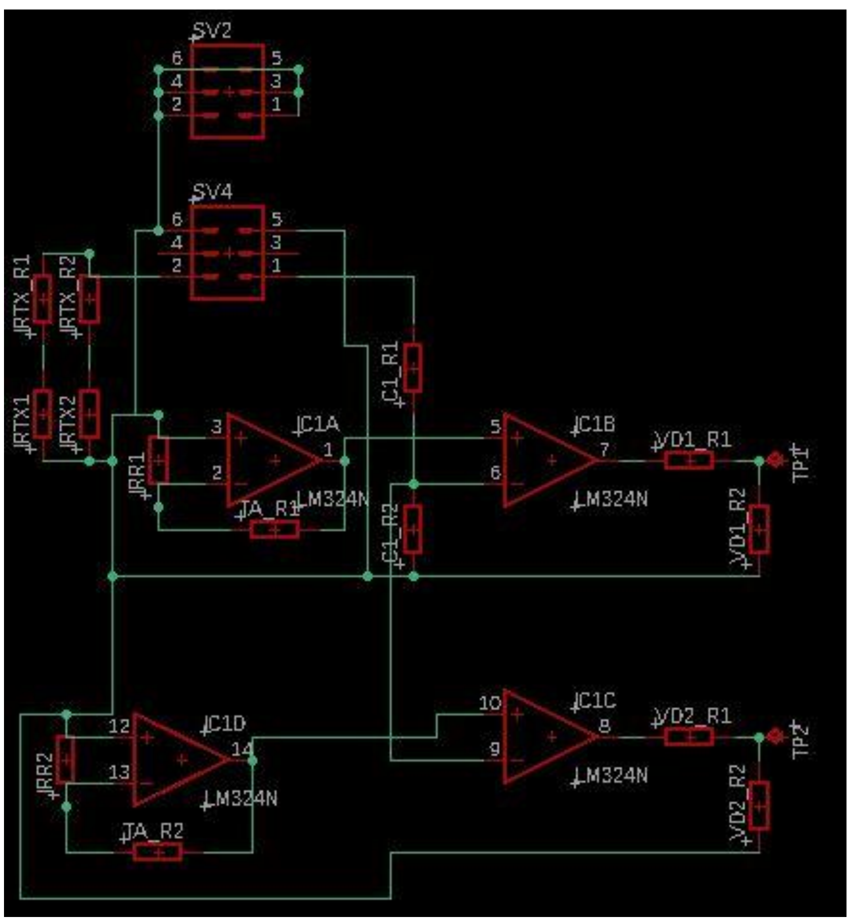

Figure 9. IR sensor schematic.

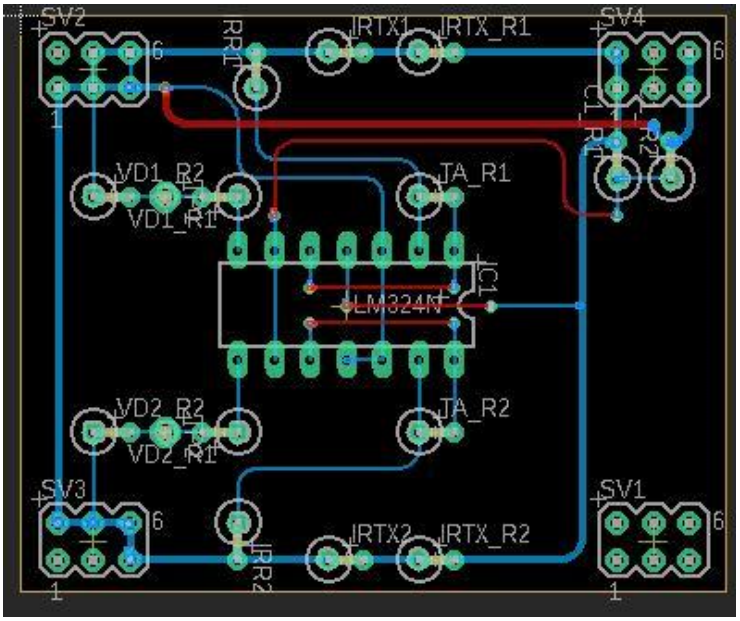

Figure 10. IR sensor PCB

In a future revision, the IR and motor control boards will be combined into a single PCB, which will further reduce the cost per robot and required assembly time.

Software Design

The software for the swarm ties the mechanical hardware and the electronics together to create the swarm. The software for the swarm was written in C and developed in the Visual Studios IDE, which was selected because of the large number of development tools it provides. The software design for this project was broken up into two main sections: subroutines and swarm algorithms.

Subroutines. Subroutines are the code to execute background processes that allow the swarm to function, such as obstacle detection, navigation, and communications. There are several other minor functions that must be handled in software, but these subroutines are the most important.

Swarm communications are achieved using the nRF24L01+ module. As noted previously, this transceiver supports an SPI link with a microcontroller, allowing it to be easily interfaced with. To use the transceiver, the microcontroller must first establish an SPI link to allow data to be sent to and from the radio. After the link is established, the microcontroller sends a set of addresses to the radio: one address is allocated as that radio's address, and the remaining addresses are those of the rest of the swarm. The microcontroller then indicates to the radio whether it should be transmitting or receiving and sends data to or receives data from the radio, depending on the selected mode. The microcontroller can also set the transmit power and Baud rate of the radio, depending on what is currently required. Implementing this in code is relatively straightforward. Short range IR communication is also simple to implement: the microcontroller broadcasts its unique address using pulse width modulation (PWM) and decodes the output of the IR detector using the same scheme. If it detects its own code, the signal is reflecting, indicating an obstacle is in the path of the robot. If it detects the address of a different robot in the swarm, it knows it is near that robot, and can update its estimated position accordingly.

The code required to implement obstacle detection is also quite simple. The main obstacle sensor is the ultrasonic sensor, which interfaces directly with the microcontroller's GPIO pins. To operate the sensor, the microcontroller would pulse the trigger pin of the sensor to generate an ultrasonic pulse and would then record the time it takes before the reflected pulse is detected. The recorded time would then be divided by two and multiplied by the speed of sound to calculate a distance to the object. Since the swarm robots can rotate in place, the robot could rotate while operating the sensor in order to generate a rough map of its surroundings.

Navigation is important for a swarm focused on target location. The code to implement basic navigation for the swarm robots relies on a mix of obstacle detection, communication with other swarm members, and the output from the IR wheel rotation sensor. By counting the frequency of the IR wheel sensor output, the microcontroller can determine the speed at which it is moving. By multiplying this speed by the time it travels at that speed, the microcontroller can determine the approximate distance traveled. A similar method can be used to generate an approximation of angle after making a turn. By communicating with other swarm members, a robot can figure out approximately which direction they traveled in and can move in their general direction using the obstacle detection code to avoid obstacles. The purpose of the navigation code is not to guide the robot to an exact location, but to bring it close enough to the target that the robot can home in on the target using short range IR communications. A flowchart of obstacle detection applied to navigation is shown in Figure 11.

Swarm algorithms. As noted in the introduction, the primary purpose of this swarm is target location, with the swarm congregating at the target after it has been located. Based on this, the utilized algorithm must cause robots to move in the direction of sensor readings indicating a possible target, and once the target is located, the algorithm must cause the swarm to travel to the target location quickly and efficiently. Based on these requirements, particle swarm optimization (PSO) was selected as the primary algorithm, with components of ant colony optimization (ACO) utilized to

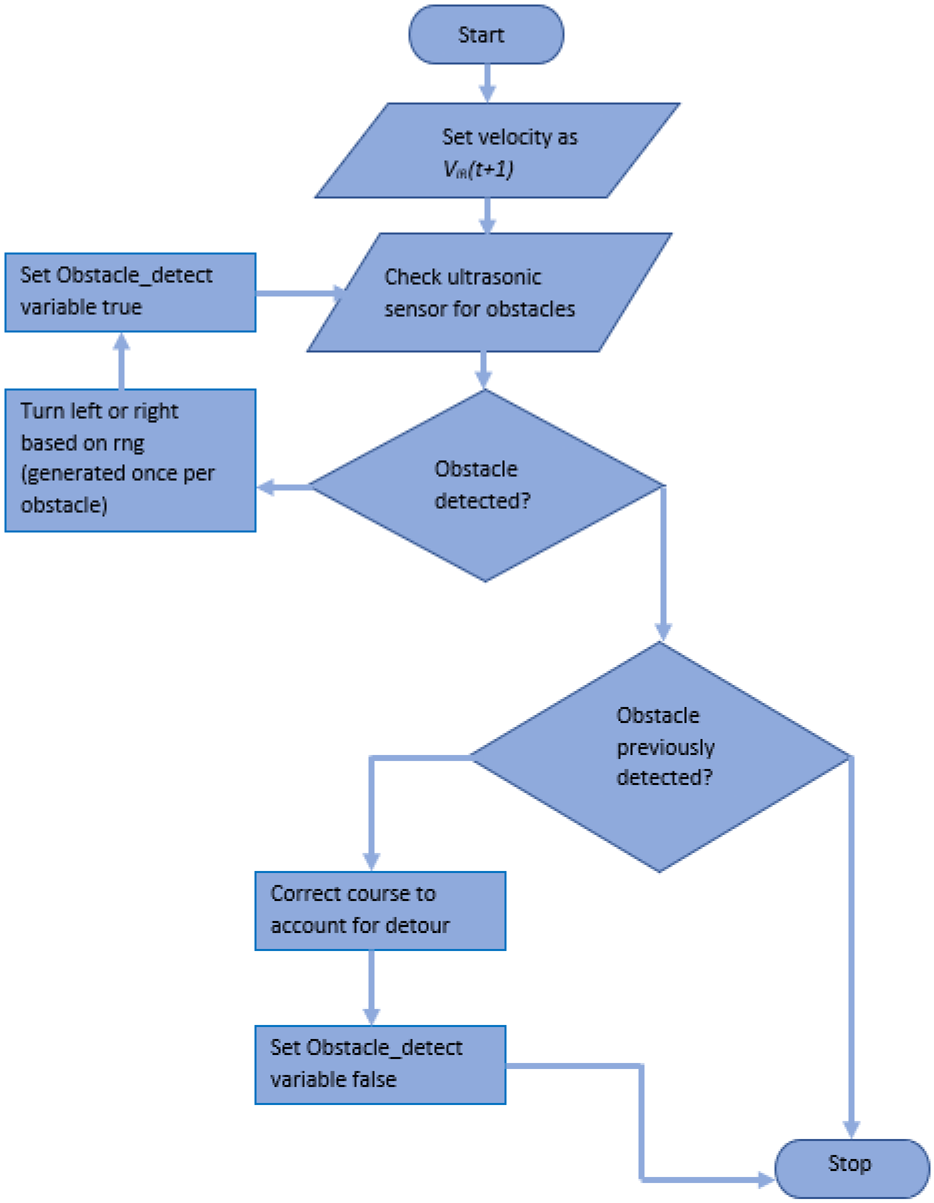

Figure 11. Obstacle detection flowchart

increase the efficiency of assembling the swarm at the target once it has been identified.

PSO algorithms are governed by each robot's individual best search result area and the entire swarm's best search result area in such a way that robots will move toward areas that match the given target parameters. One weakness of PSO algorithms is their tendency to cause a swarm to converge on a location that meets more of the target requirements than other identified locations but is still not the target location. This tendency can be lessened by selecting a proper inertial coefficient to encourage greater exploration of the search area before convergence (Ab Wahab et al., 2015).

ACO algorithms can be used to find the most efficient routes between two points and as such can be used to direct robots to the target once it has been identified. On a basic level, ACO works by directing robots down paths that have been taken by other robots previously. The more robots that have traveled down the path and the more recently this travel occurred, the greater the odds that the robot in question will follow that path (Ab Wahab et al., 2015). This characteristic of ACO can be applied to this project to direct robots toward the target once another robot in the swarm has located it.

Based on this information, the selected algorithm will be a PSO implementation with a high inertial coefficient for target location, with a simplified ACO implementation for routing swarm members to the discovered target.

Mathematical model. A PSO implementation for target location is modeled mathematically as follows:

$$V_{IR}(t+1) = w \times V_{IR}(t) + C_1 \times r_1 \times \left[X_{IR_{\text{best}}}(t) - X_{IR}(t)\right] + C_2 \times r_2 \times [X_{S\text{best}}(t) - X_{IR}(t)] \tag{1}$$

In the above equation, *w* is the inertial coefficient, which governs the relative weight of an individual robot's current velocity (V_{IR}) in determining its future velocity. Higher values of *w* result in more exploration and decrease the chances of a false convergence. C_1, the cognitive coefficient, and r_1, a randomly generated number, collectively determine the weight of the personal best position of each robot ($X_{Sbest}(t)$) in determining the future velocity of the robot. The last term of the equation contributes to the robot's velocity based on the best position located by the swarm as a whole. The magnitude of the step toward this best location is scaled by C_2 (the social coefficient) and r_2 (a randomly generated number). Both r_1 and r_2 are randomly generated numbers between 0 and 1, while C_1 and C_2 are set close to 2, and altered as needed to improve the behavior of the swarm (Ab Wahab et al., 2015). A flowchart of the algorithm is shown in Figure 12.

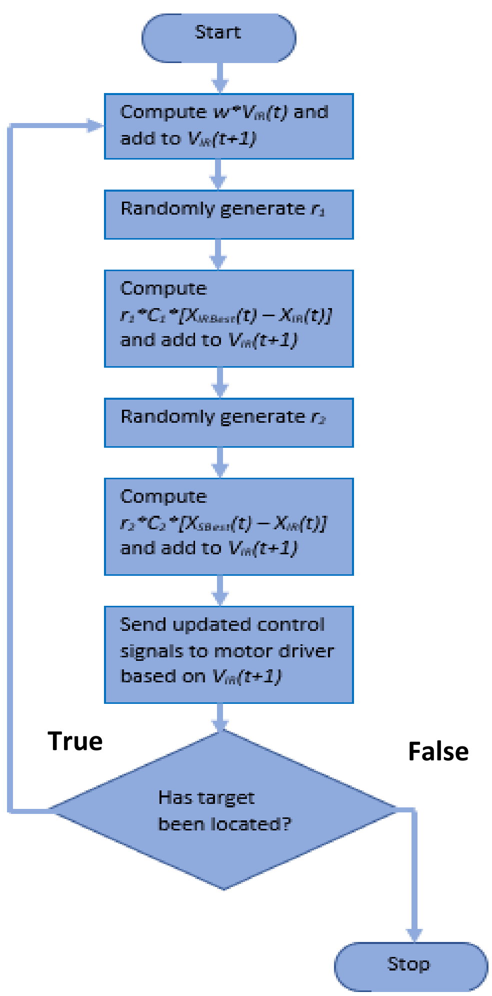

Figure 12. PSO flowchart

Budget

Table 1 shows the Bill of Materials (BOM) for the construction of a single robot following the design laid out in earlier sections of this paper:

Table 1

Bill of Materials for a Single Swarm Robot

Arduino Nano	Aliexpress	$ 1.92	1	$	1.92
L293D	Aliexpress	$ 0.26	1	$	0.26
Wheels	Banggood	$ 1.49	1	$	1.49
Motors	Banggood	$ 1.97	2	$	3.94
DIP Socket	Tayda Electronics	$ 0.05	1	$	0.05
DIP Socket	Tayda Electronics	$ 0.06	1	$	0.06
Pin Headers -F	Tayda Electronics	$ 0.29	1	$	0.29
Pin Headers -M	Tayda Electronics	$ 0.21	1	$	0.21
100u Capacitor	Tayda Electronics	$ 0.02	1	$	0.02
Resistors	Tayda Electronics	$ 0.01	20	$	0.20
1N4007	Tayda Electronics	$ 0.02	2	$	0.04
IR LED	Tayda Electronics	$ 0.06	3	$	0.18
IR Phototransistor	Tayda Electronics	$ 0.23	3	$	0.69
Prototyping Board	Tayda Electronics	$ 0.66	2	$	1.32
LM324N	Tayda Electronics	$ 0.19	1	$	0.19
NRF24L01+	Tayda Electronics	$ 0.99	1	$	0.99
Thermistor	Tayda Electronics	$ 0.12	1	$	0.12
			Total for robot:	$	13.73

As shown, the total materials cost for a single robot comes out to $13.73, not including the HDF for the body (HDF can be cheaply obtained at various hardware stores). This price can be decreased if components are purchased in larger amounts, as would be required if one were trying to create a swarm following this design. The DIP sockets are not necessary for functionality but using them decreases the risk of damaging the ICs during assembly. As expected, the majority of the cost comes from the microcontroller, the motors/gearboxes, and the wheels. Finding an equivalent microcontroller for a lower cost is unlikely, as is finding an equivalent motor/gearbox combo, so the best choice for reducing costs would be to find a replacement for the wheels, either the side-mounted wheels or the base-mounted ball bearing. 3D printing could be an option for replacing the side wheels as long as a suitable grip could be created. The ball bearing could be replaced with an acorn nut or similar smooth metal object, with a slight decrease in efficiency.

Testing

During testing, the first version of the IR wheel rotation sensor was found to be unsatisfactory. It utilized fewer components in its design than the current model in an attempt to reduce costs, but was unable to provide the desired accuracy, leading to its replacement by the current model.

A bug was also encountered with the object detection response code. When an obstacle was detected in front of a robot within approximately 3” of the sensor, the robot would stop and begin shaking rapidly. This was determined to be a timing issue between the obstacle detection and navigation code and was resolved quickly.

Conclusion

The robots created following this design are relatively low cost (<$15.00 apiece) and do not require significant assembly time. During testing, robots following this design demonstrated great maneuverability and obstacle detection, and RF communications proved to be reliable. As noted in the section discussing budget, the materials cost per robot could likely be reduced, which would make this swarm implementation even more feasible.

Next Steps

Going forwards, there are several steps to continue this project. First, more research must be performed investigating practical applications of swarm technology. Once possible applications are determined, research can be tailored to customize swarm development to best fit the selected application. A second step is continued research into developing and customizing swarm algorithms to best fit the selected application, in order to increase the intelligence of the swarm. A final step to continuing this research is to work toward a functional short-range navigation system that does not rely on GPS. This is important, since GPS is not functional in many locations (i.e., underground, underwater, inside certain buildings, etc.), and is also not accurate to very small distances, which may be required in certain circumstances.

Applications of Research

The purpose of this project was twofold: design a low-cost swarm robot that is able to interact with its environment on a larger scale than other swarm projects have done for financial reasons, and experiment with using swarm robotics for target location. The information learned from this project could be applied toward developing more rugged swarm robots and better swarm algorithm implementations for applications in the real world. Several real-world applications of this updated technology could include search and rescue operations, exploration and mapping, and surveillance. Robots have the advantage of being able to operate in hostile environments where people cannot, and a robotic swarm would be able to apply swarm intelligence toward solving problems that more traditional programming could not address as easily, giving this technology great potential. Swarm intelligence is a very powerful tool and offers great potential for application to robotic systems. There are many possible applications of swarm algorithms and swarm robotics to modern problems, and with more research and experimentation, there are endless possibilities.

References

Ab Wahab, M., Nefti-Meziani, S., & Atyabi, A. (2015). A comprehensive review of swarm optimization algorithms. *PLoS ONE, 10*(5), 1-36. doi: 10.1371/journal.pone.0122827

Arduino. (n.d.). *Arduino Nano.* Retrieved from https://store.arduino.cc/usa/arduino-nano

Ben-Gal, I., & Kagan, E. (2015). *Search and foraging: Individual motion and swarm dynamics.* New York, NY: CRC Press.

Blum, C., & Merkle, D. (2008). *Swarm intelligence: Introduction and applications.* Berlin, Germany: Springer.

Dehuri, S., Jagadev, A. K., & Panda, M. (2015). *Multi-objective swarm intelligence: Theoretical advances and applications.* Berlin, Germany: Springer.

Floreano, D., & Mattiussi, C. (2008). *Bio-inspired artificial intelligence: Theories, methods, and technologies.* Cambridge, MA: MIT Press.

Hassanien, A. E., & Emary, E. (2016). *Swarm intelligence: Principles, advances, and applications.* Boca Raton, FL: CRC Press.

Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). *Swarm intelligence.* San Diego, CA: Academic Press.

Kernbach, S. (2013). *Handbook of collective robotics: Fundamentals and challenges.* Boca Raton, FL: CRC Press.

Levi, P., & Kernbach, S. (2010). *Symbiotic multi-robot organisms: Reliability, adaptability, evolution.* Berlin, Germany: Springer.

Lim, C., Jain, L., & Dehuri, S. (2009). *Innovations in swarm intelligence.* Berlin, Germany: Springer.

Nordic Semiconductor. (2008). *nRF24L01+ single chip 2.4 GHz transceiver preliminary product specification v1.0.* Retrieved from https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf

Perry, C. (2014, August 14). The 1,000-robot swarm. *The Harvard Gazette.* Retrieved from https://news.harvard.edu/gazette/story/2014/08/the-1000-robot-swarm/

Spears, D. F., & Spears, W. M. (2011). *Physicomimetics: Physics-based swarm intelligence.* Berlin, Germany: Springer.

Sun, J., Lai, C.-H., & Wu, X.-J. (2012). *Particle swarm optimization: Classical and quantum perspectives.* Boca Raton, FL: CRC Press.

Trianni, V. (2008). *Evolutionary swarm robotics: Evolving self-organising behaviours in groups of autonomous robots* . Berlin, Germany: Springer.

Vlahavas, I., & Vrakas, D. (2008). *Artificial intelligence for advanced problem solving techniques.* Hershey, PA: IGI Global.

Warwick, K. (2012). *Artificial intelligence: The basics.* New York, NY: Routledge.

Winfield, A. (2012). *Robotics: A very short introduction.* Oxford, England: Oxford University Press.

Yang, X.-S., Cui, Z., Xiao, R., Gandomi, A. H., & Karamanoglu, M. (2013). *Swarm intelligence and bio-inspired computation: Theory and applications.* Waltham, MA: Elsevier.